Hand Motion Recognition

K.D. Yesugade, Sheetal Salunke, Ketaki Shinde, Snehal Gaikwad, Meenakshi Shingare

Abstract— Hand gesture recognition system can be used for interfacing between computer and human using hand gesture. This work presents a technique for a human computer interface through hand gesture recognition that is able to recognize 25 frams per second. The motive for designing this project is to recognize dynamic hand motions in runtime without using any external hardware. The key features of this project are customizable palm motion and backward compatibility within low cost. There is no need of any high definition camera or any costly hardware, used for recognition of hand gestures. A simple low resolution VGA camera is used. An attempt is made to use simple algorithms and to provide more user friendly environment.

This application focuses on the building block and key issues to be considered in HGRS and our contribution in the development of HGRS. The primary goal of the project is to create a system that can identify human dynamic hand gestures and use it for performing different functionalities.

In this project, an attempt is made to building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (Graphical User Interfaces), which still limit the majority of input to keyboard and mouse. Hand Gesture Recognition System enables humans to interface with the machine (HMI) and interact naturally without any mechanical devices. This could potentially make conventional input devices such as mouse, Keyboards and even touch-screens redundant.

Index Terms—Customizable Palm Motion, Backward Compatibility, HGRS.

I. INTRODUCTION

This project will design and build a man-machine inter-face using a web camera to interpret the hand gestures made by the user. The keyboard and mouse are currently the main interfaces between man and computer. In other areas where 3D information is required, such as computer games, robotics and design, other mechanical devices such as roller-balls, joysticks and data-gloves are used. Humans communicate mainly by vision and sound, therefore, a man-machine interface would be more intuitive if it made greater use of vision and audio recognition. Another advantage is that the user not only can communicate from a distance, but no need to have physical contact with the computer. However, unlike audio commands, a visual system would be preferable in noisy environments or in situations where sound would cause a disturbance. The visual system chosen was the recognition of hand gestures.

Manuscript received April, 2014.

Prof. K.D. Yesugade Computer Science, Pune University, Bharati Vidhyapeeth Collage of Engineering for Women's (BVCOEW), Pune, India.

Sheetal G. Salunke, Computer Science, Pune University, Bharati Vidhyapeeth Collage of Engineering for Women's (BVCOEW), Pune, India.

Ketaki A. Shinde, Computer Science, Pune University, Bharati Vidhyapeeth Collage of Engineering for Women's (BVCOEW), Pune, India.

Snehal C. Gaikwad, Computer Science, Pune University, Bharati Vidhyapeeth Collage of Engineering for Women's (BVCOEW), Pune, India.

Meenakshi M. Shingare, Computer Science, Pune University, Bharati Vidhyapeeth Collage of Engineering for Women's (BVCOEW), Pune, India.

The amount of computation required to process hand gestures is much greater than that of the mechanical devices; however standard desktop computers are now quick enough to make this project —hand gesture recognition using computer vision. A gesture recognition system could be used in any of the following areas:

- Man-machine interface: using hand gestures to control the computer mouse and/or keyboard functions. An example of this, which has been implemented in this project, controls various keyboard and mouse functions using gestures alone.
- 3D animation: Rapid and simple conversion of hand movements into 3D computer space for the purposes of computer animation.
- Visualization: Just as objects can be visually examined by rotating them with the hand, so it would be advantageous if virtual 3D objects (displayed on the computer screen) could be manipulated by rotating the hand in space [Bretzner & Lindeberg,1998].
- Computer games: Using the hand to interact with computer games would be more natural for many applications.
- Control of mechanical systems (such as robotics): using the hand to remotely control a manipulator

II. RELATED WORK

To improve the interaction in qualitative terms in dynamic environment it is desired that means of interaction should be as ordinary and natural as possible. Gestures, especially expressed by hands have become a popular means of human computer interface now days [3]. Human hand gestures may be defined as a set of permutation generated by actions of the hand and arm [4]. These movements may include the simple action of pointing by finger to more complex ones that are used for communication among people. Thus the adoption of hand, particularly the palm and fingers as the means of input devices sufficiently lower the technological barrier in the interaction between the disinterested users and computer in the course of human computer interaction [2]. This presents a very natural way of removing technological barriers while we are adopting the hands themselves as input devices. This needs the capability to understand human patterns without the requirement of contact sensors. The problem is that, the applications need to rely on external devices that are able to capture the gestures and convert them into input. For this the usage of a video camera can be done that grabs user's gesture, along with that we require processing system that capture the useful features and partitions the behavior into appropriate classes.

Various applications designed for gesture recognition require restricted background, set of gesture command and a camera for capturing images. Numerous applications related to gesture recognition have been designed for presenting, pointing, virtual workbenches, VR etc. Gesture input can be categorized into different categories depending on various

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

characteristic [5]. Chai et al. [6] presents a hand gesture application in gallery browsing 3D depth data analysis method. It adds up the global structure information with the local texture variation in the gesture framework designed. Pavlovic et al. [4] have concluded in their paper that the gestures performed by users must be logically explainable for designing a good human computer interface. The current technologies for gesture recognition are not in a state of providing acceptable solutions to the problems stated above. One of the major challenges is evolution in the due course of time of the complexity and robustness associated with the analysis and evaluation for gestures recognition. Different researchers have proposed and implemented different pragmatic techniques for gesture as the input for human computer interfaces. Dias et al. [7] presents a free-hand gesture user interface which is based on finding the flight of fiduciary color markers connected to the user's fingers.

The model used for the video presentation, is grounded in its disintegration in a sequence of frames or filmstrip. Liu and Lovell [8], proposed an interesting technique for real time tracking of hand capturing gestures through a web camera and Intel Pentium based personal computer. The proposed technique is implemented without any use of sophisticated image processing algorithms and hardware. Atia et al. [9] designs a tilting interface for remote and quick interactions for controlling the directions in an application in ubiquitous environment. It uses coin sized3D accelerometer sensor for manipulating the application. Controlling VLC media player using hand gesture recognition is done in real time environment using vision based techniques [10]. Xu et al. [11] used contact based devices like accelerometer and EMG sensors for controlling virtual games. Conci et al. [12] designs an interactive virtual blackboard by using video processing and gesture recognition engine for giving commands, writing and manipulating objects on a projected visual interface. Lee et al. [13] developed a Virtual Office Environment System (VOES), in which avatar is used navigate and interact with other participants. For controlling the avatar motion in the system a continuous hand gesture system is designed which uses state automata to segment continuous hand gesture and to remove meaningless motion. Xu et al. [14] presents a hand gesture recognition system for a virtual Rubik's Cube game control that is controlled by EMG and 3D accelerometer to provide a user-friendly interaction between human and computers. In this the signals segments the meaningful gestures from the stream of EMG signal inputs.

There are several studies on the hand movements especially gestures, by modeling the human body .On the basis of body of knowledge now it is possible to countenance the problem from a mathematical viewpoint [15]. The major drawbacks of such techniques are they are very complex and highly sophisticated for developing an actionable procedure to make the necessary jigs and tools for any typical application scenarios. This problem can be overcome by pattern recognition methods having lower hardware and computational overhead. These aspects have been considered in subsequent sections, by making the dynamic user interface for the validation of those concepts, where a user performs actions that generates an executable commands in an intelligent system to implement the user requirements in a natural way.

III. PROPOSED FRAMEWORK

Multiple Hand Gesture Recognition System for Human Computer Interaction aspects for implementation for the theoretical concepts. One of the key contributions of the frameworks is its capacity to enable relationships to be formed for the individual elements of the framework. Each parameter setting of the framework will have an effect on the others, and the relationships that are formed based on those settings supports a more informed process. The present research effort aims to provide a practical framework for design and development of a real time gesture recognition system for varied applications in the domain of human computer interaction. The approach is to propose a methodological approach to designing gesture interactions.

The diagram shows the fundamental relationship structure that can be drawn from applying the framework to individual systems for specific applications. There are two levels of relationships building that the framework supports; a specific level i.e. outer circle, where individual systems and the relationships between the parameter settings can inform design, and a general level i.e. inner circle where existing HCI theories and methods can be incorporated into the framework for more general applications to designing gesture systems. An effective vision based gesture recognition system for human computer interaction must accomplish two main tasks. First the position and orientation of the hand in 3D space must be determined in each frame. Second, the hand and its pose must be recognized and classified to provide the interface with information on actions required i.e. the hand must be tracked within the work volume to give positioning information to the interface, and gestures must be recognized to present the meaning behind the movements to the interface. Due to the nature of the hand and its many degrees of freedom, these are not insignificant tasks. Additionally, these tasks must be executed as quickly as possible in order to obtain a system that runs at close frame rate. The system should also be robust so that tracking can be re-established automatically if lost or if the hand moves momentarily out of the working area. In order to accomplish these tasks, our gesture recognition system follows the following architecture as shown: The system architecture as shown in figure 1 uses an integrated approach for hand gesture recognition system. It recognizes dynamic hand gestures.

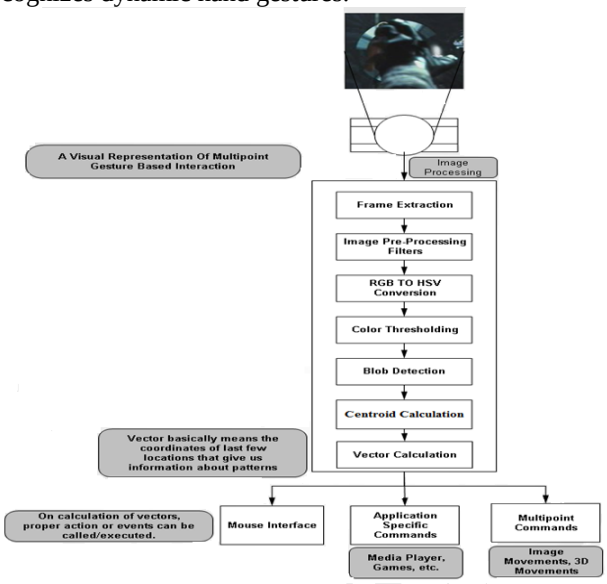

A Visual Representation Of Multipoint Gesture Based Interaction
Image Processing
Frame Extraction
Image Pre-Processing Filters
RGB TO HSV Conversion
Color Thresholding
Blob Detection
Centroid Calculation
Vector Calculation
Vector basically means the coordinates of last few locations that give us information about patterns
On calculation of vectors, proper action or events can be called/executed.
Mouse Interface
Application Specific Commands
Media Player, Games, etc.
Multipoint Commands
Image Movements, 3D Movements

Fig. - Fundamental relationship structure

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

IV. ARCHITECTURE DESIGN

The implemented system architecture starts with the background subtraction of the image which is captured by webcam in the form of videos. For background subtraction and hand detection, various image processing algorithms are used. After detection of hand feature extraction is carried out and required actions are performed.

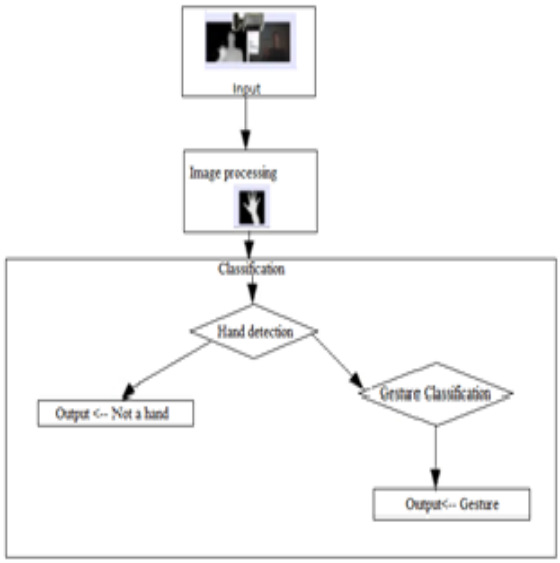


Fig.–Architecture Design

V. IMPLEMENTATION

1. BLURRING MODULE

Blurring of image is done to reduce the noise in the image. Image editors may feature a number of algorithms which can add or remove noise in an image. Some JPEG artifacts can be removed; dust and scratches can be removed and an image can be de-speckled. Noise reduction merely estimates the state of the scene without the noise and is not a substitute for obtaining a "cleaner" image. Excessive noise reduction leads to a loss of detail, and its application is hence subject to a trade-off between the undesirability of the noise itself and that of the reduction artifacts.

Noise tends to invade images when pictures are taken in low light settings. A new picture can be given an 'antiqued' effect by adding uniform monochrome noise.

BLURRING AN IMAGE:

How do we Blur an image?

Steps / Algorithm

1. Traverse through entire input image array.
2. Read individual pixel color value (24-bit).
3. Split the color value into individual R, G and B 8-bit values.
4. Calculate the RGB average of surrounding pixels and assign this average value to it.
5. Repeat the above step for each pixel.
6. Store the new value at same location in output image.

iblur= fblur(i)

In this algorithm we scan each pixel of the image linearly. Hence this algorithm is deterministic.

Sum R=r/25
Sum G=g/25
Sum B=b/25
Time complexity is O(n)

2. RGB TO HSV CONVERSION MODULE

HSV color model is preferred over RGB model. Hence RGB to HSV conversion is done. RGB has to do with "implementation details" regarding the way RGB displays color, and HSV has to do with the "actual color" components. Color is a perception based on electromagnetic waves. Natural properties of these waves are ,for example intensity and frequency. If we sweep the frequency of a light wave from infra-red to ultra-violet, we would visually perceive a color variation along the rainbow colors. Rainbow colors could be considered "pure colors" because they are represented by single-frequency waves.

Now the human eye can only respond, or "resonate" to three main light frequencies, not surprisingly red, green and blue. The fact is that this response is non-linear, so the retina can distinguish a given pure color (and implicitly its "frequency") by the combined response of the three color components.

The RGB color space exists as such only to mimic the internal workings of our retina, so that a vast majority of colors can be represented on computer displays by means of a convenient (from a computer point of view) 24 bits-per-pixel color coding. The RGB color space has no intrinsic relation to the natural color properties, neither to human interpretation of color.

For example, any arithmetical operation performed channel-wise in RGB space (for example, generation of color gradients) gives very crude or even plainly "wrong" results. That's why it is advised to create color maps by converting the color stops from RGB to other color spaces (HLS, Lab, etc.), performing the interpolations, and then converting the interpolated values back to RGB.

Algorithm:

1. Load image.
2. Read each pixel from image.
3. Separate RGB color for each pixel.
4. R = col & 0xff;
5. G = (col >> 8) & 0xff;
6. B= (col >> 16) & 0xff;
7. Find minimum value and maximum value from R, G, B.
8. Assign max to value.
9. If value equal to zero the assign hue=saturation =0.
10. Set pixel in image again.
    Else

//Formula for finding Saturation.
Find saturation= 255 * (Max-Min)/value.
if saturation is zero then
assign hue is zero.
set pixel
end if
Else
if max equal to R then

//Formula for finding Hue.
hue = 0 + 43*(G-B)/(max-min).
End if.
If max is equal to G then
hue = 85 + 43*(B-R)/(max-min).
End if
If max is equal to B then
hue = 171 + 43*(B-R)/(max-min).
End if.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

If hue<0 then
hue=hue+255.
End if.
End if.
- Set each pixel again on image.

end.

{ih,is,iv}=fhsv(iblur)

In this algorithm we convert RGB model to HSV. We scan each pixel of the image linearly. Hence this algorithm is deterministic.

Time complexity is O(n)

3. THRESHOLDING WITH HSV MODULE

Thresholding is used to create binary images. The Threshold tool transforms the current layer or the selection into a black and white image, where white pixels represent the pixels of the image whose Value is in the threshold range, and black pixels represent pixels with Value out of the threshold range. You can use it to enhance a black and white image (a scanned text for example) or to create selection masks.

Steps for thresholding:

1. Traverse through entire input image array.
2. Read individual pixel color value (24-bit) and convert it into grayscale.
3. Calculate the binary output pixel value (black or white) based on current threshold.
4. Store the new value at same location in output image.
5. Thresholding an image means converting a grayscale image into black and white (binary) image.
6. It is necessary to differentiate between foreground image and background image.

Algorithm:

1. Threshold Value (TH)=128 (This value can be changed depending on light intensity).
2. Compare grayscale value (GS) with threshold value (TH).
3. If (GS < TH) { pix = 0; // pure black}
4. else { pix = 0xFFFFFF; // pure white}

A threshold image requires 1 bit memory only.

ith = fth(i)

In this algorithm the input file is in the form of HSV . We scan each pixel of the image linearly. Hence this algorithm is deterministic

Time complexity is O(n)

4. WEB-CAM TEST MODULE

This module depends on the selection made by the user. If the Mouse Control mode is selected, the software will utilize the geometry conversion module to translate hand coordinates to the correct position on the screen.

5. BLOB DETECTION MODULE

The hand detection module scans through the image processed frames and distinguishes the different blob (Objects) detected. This method is called image segmentation. Once each blob is assigned a number, the software can then decide which blob is likely to be more similar to a hand based on its size.

What is a good blob detector?

- A filter that has versions at multiple scales.
- The biggest response should be when the filter has the same location and scale as the blob.
- In this algorithm we detect the pointer.
- Time complexity is O(n)

Algorithm :

{x,y}=fblob(ibin)

Where:

x1,x2,x3……xn belongs toX

y1,y2,y3…….yn blongs to Y

and X, Y are set of N(x,y) co-ordinates of a pixel.

Now,

ai= M(X, Y, T)

t1,t2……tn belongs toT (set of gesture templets)

ai = action to be processed.

VI. APPLICATIONS

1) Gaming:

Here we can use the Hand Gesture Recognition for Gaming. For the gaming we do not have to redesign the whole system. We can play any game like solitaire, NFS gaming etc.

2) Media player: (Winamp)

Here we can use the Hand Gesture Recognition for Media Player. In media layer control we can change the song which is currently playing using hand gestures and we can increase or decrease the volume of media player.

3) O.S control:

Here we can use the Hand Gesture Recognition for O.S Control. We can do any normal functions which we regularly do on Pc.

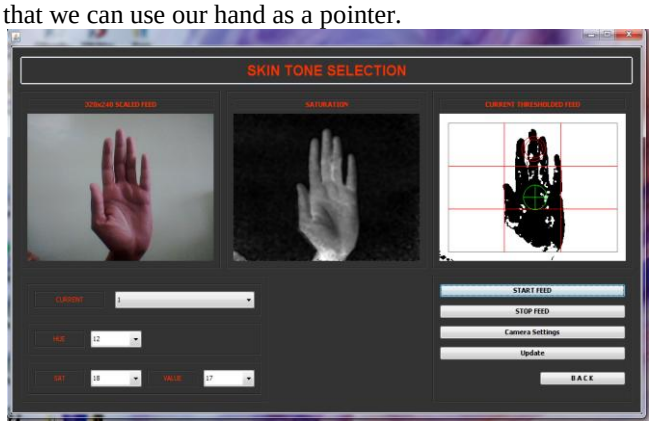
VII. RESULT

We carried out experiments with users to evaluate the functionality of the proposed gesture-detection algorithm. We requested the users to carry out a series of gestures in front of the webcam and measured whether the detection was successful. An observer recorded whether the output produced by the algorithm corresponded to the actual gesture being made. The lighting, camera position, and image background were controlled,

The user was shown a gesture sequence—on a computer screen. Each gesture sequence contains a randomly permuted sequence of hand gestures to perform. The sequence was available on the screen while the user performed the gestures. When the user finished performing the last gesture in the sequence, a new random sequence was show.

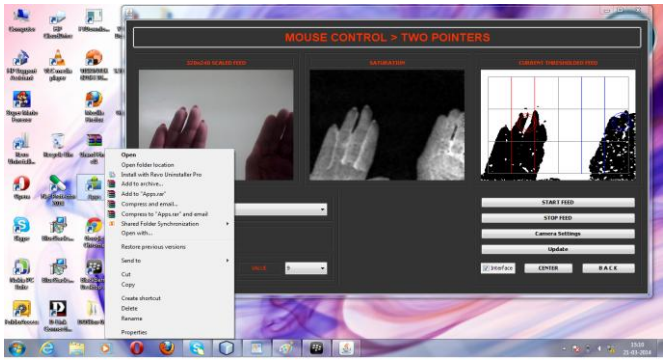
Snap Shots:

As we are using customizable pointer selection, to operate application with bare hand we need to select the skin tone so that we can use our hand as a pointer.



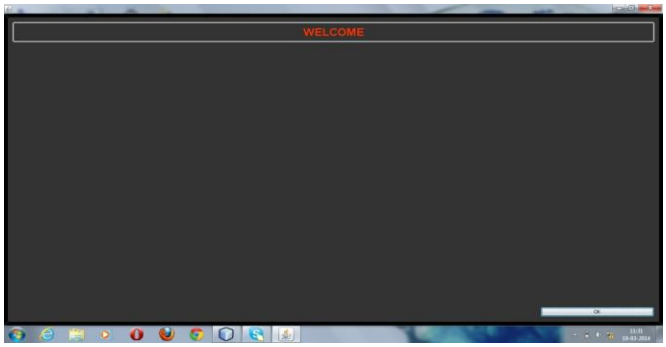
Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

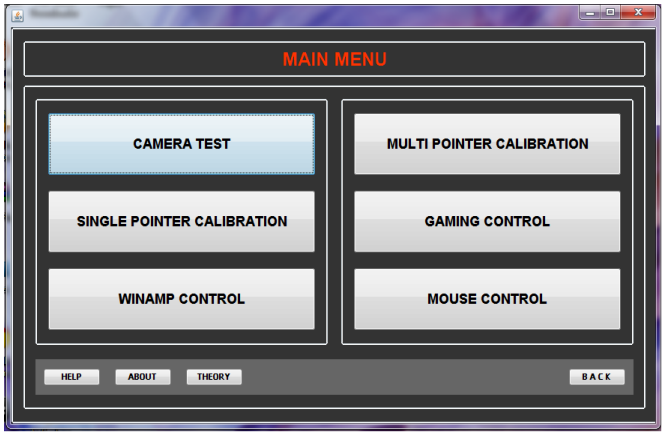
Here we are using two pointers for controlling the OS. Left hand is used for navigation of the mouse pointer and right hand is used for performing functions such as left click, right click, selecting an object and dragging



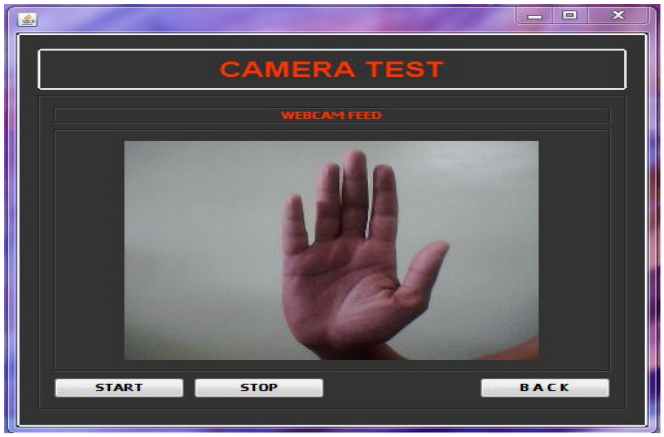
VIII. USER MANUAL

- The connection webcam and system should be checked frequently.



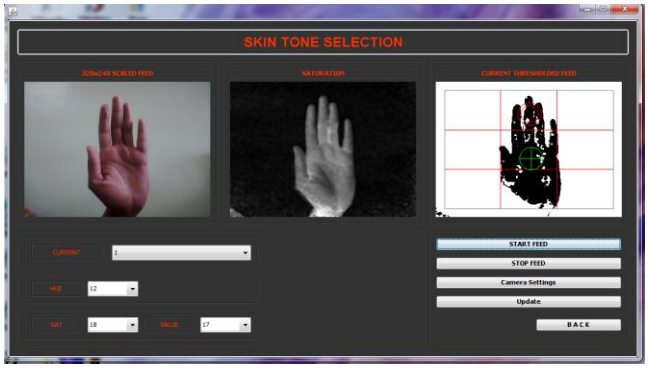


- User must first select camera test for testing the working of camera.



- **Single pointer Calibration:**

In this user is able to select skin tone or any other color as an input pointer just by clicking on that color in the frame. Green pointer is the centroid of the blob and red pointer is the highest point of the blob.

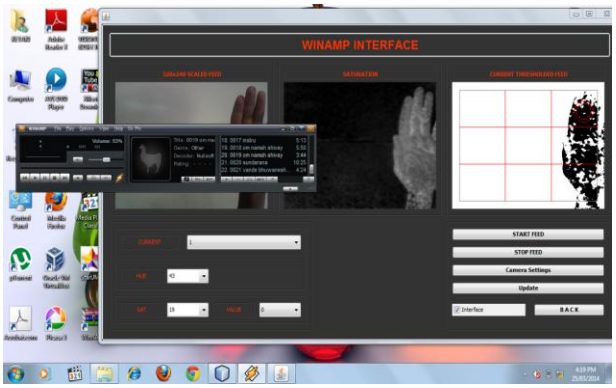


- **Winamp Control:**

We use single pointer for controlling the winamp. There are nine quadrants on screen.

1	**2**	**3**
4	**5**	**6**
7	**8**	**9**

Functions of quadrants:
Quadrant 1: Is used for decreasing the volume.
Quadrant 2: Pause/ Play
Quadrant 3: Decreasing the volume.
Quadrant 4: Previous track.
Quadrant 5: To rest the pointer
Quadrant 6: Next track.
Quadrant 7, 8, 9: No function.



- **Multi-pointer Calibration:** We are using two pointers for controlling the mouse.
  - Here there are two quadrants one for each hand.

L1	L2	L3	R1	R2	R3
L4	L5	L6	R4	R5	R6
L7	L8	L9	R7	R8	R9

  - L1- L9 are used for left pointer.
  - R1-R9 are used for right pointer.

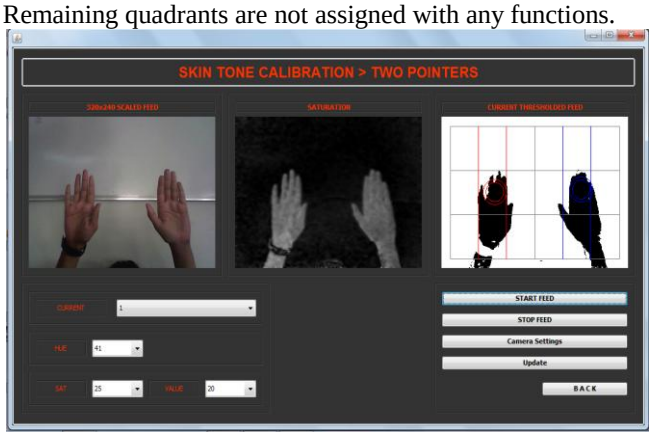
Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

- Left hand is used for navigation of the mouse pointer and right hand is used for performing functions such as left click, right click, selecting an object and dragging.

Functions of Quadrants

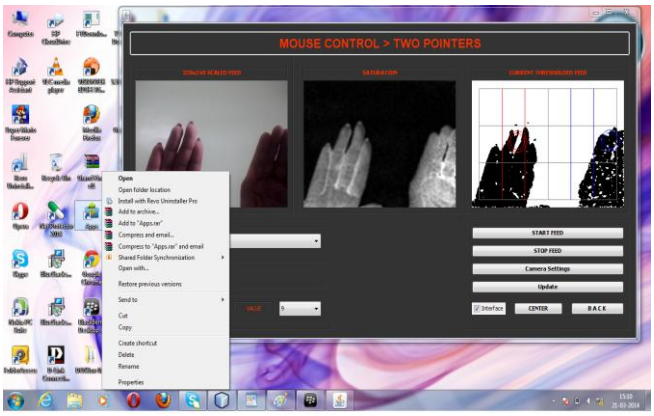
Quadrant L2: Moving the pointer towards upside.
Quadrant L4: Moving the pointer towards left side.
Quadrant L6: Moving the pointer towards right side.
Quadrant L8: Moving the pointer down.
Quadrant R3 and R1: Used for selecting and releasing the object.
Quadrant R4: Left click.
Quadrant R5: Right click.

Remaining quadrants are not assigned with any functions.



➢ **Mouse Control:**

We are using two pointers for controlling the mouse.



IX. CONCLUSION

After analyzing the existing model of hand gesture recognition, we concluded that we could add some more functionality to the system regarding pointer selection and background requirements on a major basis. This helped us to insert few new features such as:

- Detecting bare hand dynamic gestures.
- Customizable Pointer Selection.
- Simultaneous two pointer detection.
- Backward Compatibility.
- Processing 25 frames /second.
- Low Cost (No need of high definition camera)

Thus our proposed model has tried to implement them.

X. FUTURE WORK

As a future work, we hope to relax the lighting, camera position, and image background requirements in future work, as the proposed method is designed to accommodate a less restricted use setting.

REFERENCES

[1] Conic, N., Cerseato, P., De Natale, F. G. B.,: Natural Human- Machine Interface using an Interactive Virtual Blackboard, In Proceeding of ICIP 2007, pp.181-184, (2007). 64 Real Time Multiple Hand Gesture Recognition System for Human Computer Interaction Copyright © 2012 MECS I.J. Intelligent Systems and Applications, 2012, 5, 56-64

[2] Vardy, J. Robinson, Li-Te Cheng, "The Wrist Cam as input device", Wearable Computers, 1999

[3] Wong Tai Man, Sun Han Qiu, Wong Kin Hong, "ThumbStick: A Novel Virtual Hand Gesture Interface", In Proceedings of the IEEE International Workshop on Robots and human Interactive Communication, 300-305.

[4] W. T., Freeman, D. B Anderson, and P. et al. Beardsley. "Computer vision for interactive computer graphics. IEEE Trans. On Computer Graphics and Applications, 18:42-53, 1998.

[5] N.Soontranon, S. Aramvith, and T. H. Chalidabhongse, "Improved face and hand tracking for sign language Recognition". IEEE Trans. On ITCC, 2:141-146, 2005.

[6] V. Pavlovic, R. Sharma and T.S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), vol. 7(19), pp. 677–695, 1997.

[7] Xiujuan Chai, Yikai Fang and Kongqiao Wang, "Robust hand gesture analysis and application in gallery browsing," In Proceeding of ICME, New York, pp. 938-94, 2009.

[8] José Miguel Salles Dias, Pedro Nande, Pedro Santos, Nuno Barata and André Correia, "Image Manipulation through Gestures," In Proceedings of AICG'04, pp. 1-8, 2004.

[9] Ayman Atia and Jiro Tanaka, "Interaction with Tilting Gestures in Ubiquitous Environments," In International Journal of UbiComp (IJU), Vol.1, No.3, 2010.

[10] [10] S.S. Rautaray and A. Agrawal,"A Novel Human Computer Interface Based On Hand Gesture Recognition Using Computer Vision Techniques," In Proceedings of ACM IITM'10, pp. 292-296, 2010.

[11] Z. Xu, C. Xiang, W. Wen-hui, Y. Ji-hai, V. Lantz and W. Kong-qiao, " Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors," In Proceedings of IUI'09, pp. 401-406, 2009.

[12] S. Lee, S. W. Ghyme, C. J. Park and K. Wohn, "The Control of avatar motion using hand gesture," In Proceeding of Virtual Reality Software and technology (VRST), pp. 59-65, 1998.

[13] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang and J. Yang, "A framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors," IEEE Trans. On Systems, Man and Cybernetics- Part A: Systems and Humans, pp. 1-13, 2011.

[14] N. Conci, P. Cerseato and F. G. B. De Natale, "Natural Human-Machine Interface using an Interactive Virtual Blackboard," In Proceeding of ICIP 2007, pp. 181-184, 2007.

[15] Yi, F. C. Harris Jr., L. Wang and Y. Yan, "Real-time natural hand gestures", In Proceedings of IEEE Computing in science and engineering, pp. 92-96, 2005.

Prof. K.D. Yesugade M.E computer from Bharati Vidhyapeeth University. Joined as assistant professor in BVCOEW in 2003

Sheetal Ghanshyam Salunke Perusing B.E. in Bharati Vidhyapeeth Collage of Engineering for Women's (BVCOEW), Katraj-43, Pune University.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

Ketaki Ajay Shinde Perusing B.E. in Bharati Vidhyapeeth Collage of Engineering for Women's (BVCOEW), Katraj-43, Pune University.

Snehal Chandrakant Gaikwad Perusing B.E. in Bharati Vidhyapeeth Collage of Engineering for Women's (BVCOEW), Katraj-43, Pune University.

Meenakshi Maruti Shingare Perusing B.E. in Bharati Vidhyapeeth Collage of Engineering for Women's (BVCOEW), Katraj-43, Pune University.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.